assignment1: this assignment builds

1. 2 regression models for calculating sepal width then compares them and prints stats for comparing models
2. 2 classification models for calculating type using quarantiles and prints stats for comparing models

In [9]:
```
# Import necessary libraries
from sklearn import datasets
import pandas as pd

iris= pd.DataFrame(datasets.load_iris().data)
iris.columns = datasets.load_iris().feature_names

iris['type'] = datasets.load_iris().target

iris['type']=iris['type'].astype('object')

# Display the first few rows of the DataFrame
print(iris.head())

iris
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

  type
0    0
1    0
2    0
3    0
4    0
```

Out[9]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	type
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [10]:
```
iris['new']=(iris.iloc[:,0]*iris.iloc[:,1])/(iris.iloc[:,2]*iris.iloc[:,3])
```

In [11]:
```
iris.iloc[:,4]
```

Out[11]:
```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: type, Length: 150, dtype: object
```

Sample 80% of the data for a training set stratifying on 'type'

In [12]:
```
from sklearn.model_selection import train_test_split as tts



X_train, X_test, y_train, y_test = tts(iris.iloc[:,0:4], iris.iloc[:,5], test_size=
```

In [13]:
```
len(X_train), len(X_test), len(y_train), len(y_test)
```

Out[13]:
```
(120, 30, 120, 30)
```

build regression models and print ME, MPE, MAE, MSE, MAPE

In [14]:
```
import numpy as np
from sklearn.metrics import mean_absolute_percentage_error as MAPE
from sklearn.metrics import mean_squared_error as MSE, mean_absolute_error as MAE

def myf(y,yhat):
    ME=np.round(np.mean(y-yhat),3)
    MPE=np.round(np.mean((y-yhat)/y),3)
    myMAE=np.round(MAE(y,yhat),3)
    myMSE=np.round(MSE(y,yhat),3)
    myMAPE=np.round(MAPE(y,yhat),3)
    print("\n","ME:", np.round(ME,3),"\n","MPE:",MPE,"\n", "MAE:",
        myMAE,"\n", "MSE:", myMSE,"\n","MAPE:",myMAPE)

est1=np.mean(X_train['petal length (cm)'])

est2=np.mean(X_train['sepal length (cm)']-X_train['petal width (cm)'])

est1=[est1]*len(y_test)

est2=[est2]*len(y_test)


myf(X_test['sepal width (cm)'],est1)

myf(X_test['sepal width (cm)'],est2)
```

```
ME: -0.677
MPE: -0.237
MAE: 0.694
MSE: 0.602
MAPE: 0.242

ME: -1.543
MPE: -0.522
MAE: 1.543
MSE: 2.526
MAPE: 0.522
```

On the test set, evaluate the two classifiers (built on the training set) below for 'type' using accuracy, precision, recall, and the F1 score.

Up to 1st quantile of sepal length = type 0, >1st up to 2d quantile = type 1, >2d quantile = type 2

Up to 2d quantile of sepal length = type 0, >2d up to 3d quantile = type 1, >3d quantile = type 2

In [15]:
```
from numpy import percentile
from sklearn.metrics import confusion_matrix as cm, ConfusionMatrixDisplay as cmd
from sklearn.metrics import classification_report as cr
import matplotlib.pyplot as plt
```

```
est3=percentile(X_train['sepal length (cm)'], [25, 50])
est4=percentile(X_train['sepal length (cm)'], [50,75])

y_hat=np.zeros(len(y_test))

y_hat[X_test['sepal length (cm)']>est3[0]]=1

y_hat[X_test['sepal length (cm)']>est3[1]]=2

y_hat=y_hat.astype('int')

print(cr(y_test.astype('int'),y_hat))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.00      0.00      0.00         7
           2       0.35      0.86      0.50         7
           3       0.00      0.00      0.00         6
          42       0.00      0.00      0.00         1
          50       0.00      0.00      0.00         1
          53       0.00      0.00      0.00         1
          54       0.00      0.00      0.00         1
          56       0.00      0.00      0.00         1
          58       0.00      0.00      0.00         1
          63       0.00      0.00      0.00         1
          66       0.00      0.00      0.00         1
          82       0.00      0.00      0.00         1
          96       0.00      0.00      0.00         1

    accuracy                           0.20        30
   macro avg       0.03      0.06      0.04        30
weighted avg       0.08      0.20      0.12        30
```

```
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Precision is ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control th
is behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Recall is ill-defined and being set to 0.
0 in labels with no true samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Precision is ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control th
is behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Recall is ill-defined and being set to 0.
0 in labels with no true samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Precision is ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control th
is behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Recall is ill-defined and being set to 0.
0 in labels with no true samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
```

In [16]:

```
y_hat2=np.zeros(len(y_test))

y_hat2[X_test['sepal length (cm)']>est4[0]]=1

y_hat2[X_test['sepal length (cm)']>est4[1]]=2

y_hat2=y_hat2.astype('int')

print(cr(y_test.astype('int'),y_hat2))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.12      0.14      0.13         7
           2       0.11      0.14      0.12         7
           3       0.00      0.00      0.00         6
          42       0.00      0.00      0.00         1
          50       0.00      0.00      0.00         1
          53       0.00      0.00      0.00         1
          54       0.00      0.00      0.00         1
          56       0.00      0.00      0.00         1
          58       0.00      0.00      0.00         1
          63       0.00      0.00      0.00         1
          66       0.00      0.00      0.00         1
          82       0.00      0.00      0.00         1
          96       0.00      0.00      0.00         1

    accuracy                           0.07        30
   macro avg       0.02      0.02      0.02        30
weighted avg       0.06      0.07      0.06        30
```

```
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Precision is ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control th
is behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Recall is ill-defined and being set to 0.
0 in labels with no true samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Precision is ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control th
is behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Recall is ill-defined and being set to 0.
0 in labels with no true samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Precision is ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control th
is behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
C:\Users\anujb\AppData\Roaming\Python\Python313\site-packages\sklearn\metrics\_class
ification.py:1731: UndefinedMetricWarning: Recall is ill-defined and being set to 0.
0 in labels with no true samples. Use `zero_division` parameter to control this beha
vior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
```

In [ ]: